Grenze International Journal of Engineering and Technology, June Issue

Log Management System

Nakesh Tewari[1], Kshitij Kumar Yadav[2], Kunal Dewangan[3], Manish Singh[4] and Puneet Goyal[5]
[1-5]Student, Computer Science, KIET Group of Institutions, Delhi-NCR, India

***Abstract*— Server and applications in current IT environments generate a continuous flow of log data, and it can be in volumes that are overwhelming. Dealing with such information manually is not feasible and this creates a need to have a system that would automate and centralize the whole process of log-management. The given project proposes a solution that could help to collect logs of different sources, convert them into a standard format, and store them in a form that will allow one to perform quick search and analysis. With the consolidated platform where all the logs are located, the administrators can monitor the system behavior, any security related activities, and understand general performance easily without the need to jump between various tools or machines. The system gives smart processing techniques, e.g. machine learning models, which can identify unusual patterns, and predict anomalies independently. This enables any impending issues to be caught at the initial stages before it impacts on users. It also features convincingly straightforward dashboards and search solutions that assist the IT groups to maintain a watch on the activity of their systems, identify errors, and trace the problems to their origins in a swift manner. Besides that, there is security and compliance in the construction of the platform. It keeps the log data encrypted and under strict access policies and assists organizations to keep the sensitive data secure and comply with the regulations. This log management is to be scaled and expanded with ease, thus it is suitable for the hybrid cloud infrastructure, classic on-premises infrastructure, and current containerized infrastructure. Its scalable design provides it with the ability to scale to meet a growing workload, as well as to integrate with new technologies as systems change.It reduces the amount of manual work and automates the routine analysis of the logs, enhances the utilization of resources, and accelerates the speed at which a team can respond to incidents. The result is that this project offers an organization with an effective tool that enhances reliability and security and offers more insight into their overall IT environment with the help of efficient management of logs.**

***Keywords*— Log Management; Centralized Logging; Anomaly Detection; Machine Learning; Real-Time Monitoring; Data Normalization; Security Compliance; System Performance; Operational Visibility; Scalability; Automation; Cloud Infrastructure.**

I. INTRODUCTION

The modern IT landscape has seen all servers, applications, and devices on the network continually produce logs - massive amounts of logs. These logs can record every form of information, including what the system is doing, how users are interacting with the system, where security vulnerabilities may be appearing, and what the performance of the entire system is like. Tracking all of this information is important if you would like your systems to remain healthy, secure and easy to troubleshoot.

Grenze ID: 01.GIJET.12.2.241_20

The purpose of this project is to come up with a centralized server system of log management, which combines logs of the various sources into one platform. The system allows real-time ingredients, normalization, filtering and safe storage of log information and offers high-quality analytics and visualization features to enable IT staff with actionable information.

The solution enables proactive monitoring, performance bottlenecks and security threats early detection through machine learning-based anomaly detection and automated alerting, and compliance requirements through encrypted and access-controlled log repositories.

The log management system will be able to scale with ease and match any type of environment that may be in a hybrid cloud environment, supporting container-based applications, or in a traditional on-premises environment. It addresses such typical issues as supporting various types of logs, eliminating irrelevant noise, efficient storage, and long-term data retention without increasing the cost.

Ultimately, the idea is to make organizations run more effectively, reduce outages and enhance their overall security by converting complex, and distributed log data into valuable and timely insights to the administrators and security teams.

This system is also important in enhancing organizational decision-making and resiliency in addition to its technical capabilities. It enables the IT administrator and security analyst to react swiftly to irregularities, security breaches, and performance problems by transforming raw, unstructured log data into actionable information. Its capability to correlate events between networks, applications and databases provides a complete picture of operational view that may otherwise be overlooked by the traditional monitoring tools. This wide visibility is able to not only support and assist in responding to incidents faster, but also facilitates capacity planning, compliance audits, as well as long-term IT resource optimization. With increasing complexity of digital infrastructures, the centralized and smart log management system will be critical in achieving high availability, organizational trust, and business continuity-based data strategies

A. Background

In the modern-day digital era, big companies rely on complex IT systems which consist of servers, applications, network devices, databases, and security systems. A combination of these elements creates huge amounts of log data in each instant. This data contains information in detail about the events in the system, user activity, transactions, errors and security notifications, which can be important information about the behavior and performance of the systems. The proper handling and analysis of these logs is imperative towards ensuring the availability of the system, prompt problem trouble shooting, security threats identification, regulatory and overall business continuity.

B. Problem Statement

E-governance systems generate high volumes of log data in their various services and applications which record the activities, transactions and system events of the users. Nevertheless, most government agencies nowadays are characterized by the practice of fragmentation and decentralization of log management, and it is hard to recognize the operational issues or security threats very fast. In the absence of a content-based classification and automated notifications, essential pieces of logs may be lost in the tidal inundation of information without notifying the key participants, slowing a response and resolution.

Centralized log management system is needed to improve the level of transparency, reliability and security of the e-governance services. Intelligent classification of log content should also be part of such a system to prioritize the critical events and create timely notifications to the administrators. By taking care of their needs, the system has the potential to enhance operation monitoring, reduce the downtime, and meet the government standards including the recommendations of AICTE.

C. Objectives

Plan an effective log management server that can effectively gather and consolidate logs on a large scale and incrementally across a broad set of distributed IT resources, such as cloud instances, on-premises servers, network equipment, and containerized systems.

Introduce automated parsing and normalization, to standardize various formats of logs, aiding easier querying, correlating and analysis of log data.

Embrace the newest analytics, such as machine learning, to identify anomalies, enable predictive maintenance, and detect any possible security threat before and Elaborate on user-friendly, customizable dashboards and reporting capabilities that can provide actionable insights and help in auditing compliance and supporting forensic investigations.

D. Challenges

Data Format Heterogeneity: Heterogeneity of Data Formats Modern IT systems have vastly divergent sources of logs, such as applications, servers, network equipment, and cloud services. All these sources tend to have various forms and formats of the logs which complicates the comparison of all this data and standardization of the information that can be meaningfully analyzed. Normalization and consolidation of such heterogeneous data are very important to successful monitoring, correlation and reporting.

E. Efficient Content-based

Classification: The development of reliable algorithms that automatically identify the content of logs to indicate vital events and filter unwanted information is a major concern.
Timely Notification: It is necessary to design an intelligent notification system that will inform the stakeholders about the significant events in a timely manner without flooding them with non-true positives.
Scalability and Performance: The ability to process and ingest logs on a large scale particularly during peak traffic or security incidents needs a scalable and efficient infrastructure.
Security and Compliance: It is essential to make sure that logs are sent, stored and accessed with security and according to the regulations to avoid losses to unauthorized access and safeguard the integrity of data.

III. PROPOSED SYSTEM

To tackle the challenges faced by current log management in e-governance systems, this project proposes a centralized server log management solution with intelligent content- based classification and real-time alerting. The primary goal is to automate the collection, categorization, and notification of log data, enabling faster operational monitoring, quicker response to security incidents, and streamlined compliance reporting across government IT environments.

II. LITERATURE REVIEW

TABLE I

Authors	Key Findings/Contributions	Limitations	Methodology/ Approach	Dataset
Ch. Yamini et al. (2023)	This project showed that artificial intelligence, or to be more precise the algorithms of Random Forest, may automate the log processing and properly categorize the anomalies and even anticipate the possible problems better than the manual or rule-based systems. AI-powered system scan handle the logs of various sources at once, producing timely and proactive notifications, and saving a lot of manual work and avoiding the human mistake.	Various sources of logs are not pre-processed in the same way, so AI analysis is not simple. Not many full-scale AI log management systems are in operation. IT operators may have problems understanding and believing in AI models.	Created a log management system that uses AI. Collected, reprocessed, and classified server logs. Comparing AI results with manual and traditional methods.	Collected logs of the servers in the cloud and on-premises to mimic the real- world processes
New Relic (2022)	Organizations are migrating towards cloud-native, centralized log management to gain scalability, real time monitoring, as well as advanced analytics. Some of the key challenges that can be identified are how to manage increasing volumes of logs, facilitate unified analytics, and offer adequate retention to maintain compliance and security.	Tool sprawl, fragmented platforms remain to be a problem. Systems that are large increase costs and complexity dramatically. Automated alerting and incident response are not yet widely implemented.	Surveys and telemetry analysis on thousands of organizations. Analyzed patterns and trends of its usage from 2022-2023.	Anonymized and aggregated logs and survey data of customer and partner SaaS platforms.
Dynatrace Engineering Team (2024)	Emphasized the importance of AI-powered, centralized log management in hyperscale and multi-cloud environments. Demonstrated how real-time	Integration of logs of microservices, on-premises systems, and cloud environment is complicated.	Real-life deployments studied to get familiar with real-life log management.	Analyzed enterprise production logs from cloud native environments for latency and

	monitoringcan predict outages, optimize resource usage, and quickly detect security incidents.	Managing hyperscale deployments are challenging both Performance and cost wise.	Defined the standards of log analysis pipelines with hyperscale workload.	operational efficiency.
Fact MR Market Report (2024)	Cloud-based log management is being rapidly adopted for scalability, flexibility, and regulatory compliance. Vendors will include new features such as root cause analysis, anomaly detection, and SOC integration.	Log management being centralized continues to raise concerns regarding data protection and privacy. The problem of cyber threats is also growing at a rate that is outpacing most of the existing monitoring systems.	Made market survey and interview with the vendors, as well as case study analysis. Reviewed regional and sector-specific adoption trends from 2022 to 2024.	Examined sales growth, adoption rates, and vendor features across global industries.
SigNoz (2024)	Centralized logging will be used to boost system visibility, troubleshooting, compliance, and security in distributed environments.	Ensuring consistent log quality and granularity remains challenging.	Evaluated five open- source centralized logging solutions. Implemented using Open Telemetry and Sigos.	Organized and indexed logs collected from both simulated cloud-native.
Alvarez & Singh (2024)	Additionally, we constructed a streaming log analytics solution with the help of Apache Kafka and Spark streaming. Reduced latency in identifying anomalies and security breaches across distributed microservices. Built-in automation of notifications using webhooks to Slack and email so that the response is done immediately.	Kafka and Spark use high costs to maintain their clusters. Set-up and management require high-level DevOps skills. Minimal usage in companies that have limited resources.	Created a Kafka-based log ingestion pipeline. Implemented the jobs of the Spark streaming to detect anomalies in real time. Measured latency, throughput, and scalability of the system.	Tested and analyzed a b o u t 100 GB of synthetic syslog dataof microservices deployment.

A. System Overview

The proposed system combines scalable log collection agents, a centralized log server, and advanced analytics modules that use machine learning for content-based classification of log events. It enables smooth ingestion of logs from multiple government departments and e- services, standardizes diverse log formats, and employs both rule- based and AI-powered classifiers to categorize events by type and severity. Critical incidents trigger real-time notifications, allowing administrators to respond promptly. The system is designed with modularity, security, and extensibility in mind, making it easy to integrate with existing e-governance platforms and support future upgrades.

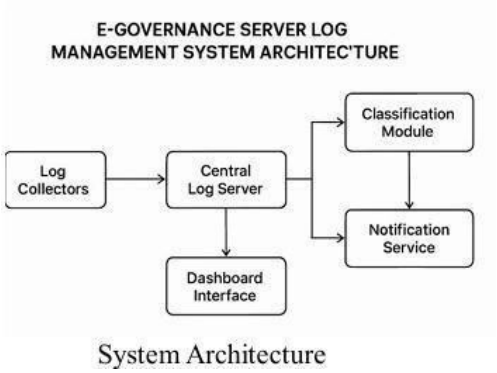

Fig. 2. Sequence Diagram of Log Flow

B. Key Components

Lastly, cross-disciplinary collaboration is a vital step to translating these high-tech methods into practical use and whether it is in healthcare applications such as early melanoma detection or in the control of complex IT systems.

- Log Collection Agent Small, lightweight Log collection agents attach to servers and other network devices and run quietly gathering log data 24/7. They can use various languages such as syslog, windows event logs or APIs and send all that information to the central hub in a secure manner.
- Centralized Log Server This is the center of the system. It collects repositories and catalogues huge quantities of logs. It renders the whole searchable, visualizable, and scalable to analyze with the help of such tools as Elasticsearch, Logstash and Kibana.
- Real-time Notification time lets administrators stay informed. The system, which is based on such platforms as Kafka or RabbitMQ, may either push the notification immediately through email, SMS, or dashboard and ensure that nothing crucial slips under the carpet.
- Dashboard and Reporting The interactive side to the user is smooth. Constructed in React, or Angular, it can show health of a system, view logs, create reports and manage alerts, all in a responsive safe interface which communicates with the backend via APIs.

C. Features and Innovations

- Multi-intent Processing The system can interpret and execute multiple user intents within a single message that would enable it to effectively process complex or mixed queries.
- API Support of Live-Data Sources The system supports dynamically active and up-to-date responses by connecting to live data sources and external APIs.
- Scalability and Security The system is built on microservices architecture and can easily increase its workloads. To ensure that the communications and user information is secure, JWT-based authentication is used to protect the data.

IV. METHODOLOGY

A. System Architecture

The server log management system consists of four main components: ->

- Log Collection Agents: Log Collection Agents: Lightweight agents that run on the servers and network devices perform real time log collection and forward that data into the central system.
- Central Log Server: This is a fundamental component that accepts, processes, structures, and stores massive amounts of different logs. It also does indexing and optimized retrieval to allow speedy and efficient queries.
- Classification and Analytics Engine: This engine is a combination of rule-based filters and machine learning models, where logs can be classified based on severity, source, and event type and anomalies and potential security incidents are detected.
- Notification and Dashboard Interface: The administrators are informed in real-time either through email or SMS notifications or system notifications. It has a friendly dashboard that enables one to monitor, search and report system logs.

B. Tech Stack

- Programming Languages: JavaScript— to be used as backend processing, analytics and integration.
- Frameworks: Node js for the backend services; Flask for lightweight API endpoints
- Logging Tools: ELK Stack (Elasticsearch, Logstash, Kibana) is an efficient tool in log ingestion, storage, search and visualization.
- Messaging and Notification: Apache Kafka or RabbitMQ can be used to support the scalable real-time streaming of messages and providing alerts.
- Database: MySQL is used to store metadata and configuration, and MongoDB is used to store unstructured log data.

C. Development Workflow

- Requirement Analysis and Log Source Identification: Establish the systems that generate logs and establish the requirements of collecting them effectively.
- Backend Development: Implement the central log server, ingestion pipelines and storage to support large amounts of log data.
- Classification and Analytics Module Development: Develop and train machine learning models of content-based logs classification and anomaly detection.

- Dashboard and Notification System Development: Introduce real time alerting systems and come up with user friendly dashboards to monitor and report.
- User Feedback and Continuous Improvement: Observe feedback and use it to make improvements to the features to increase usability and system performance in general.

D. Implementation

- Domain-Specific Log Collection and Ingestion: The role of domain-specific agents is to collect logs on numerous servers and e-governance applications across government. These agents are configured to collect data in a similar and complete manner in distributed environments by supporting various sources and formats.
- Automated Log Parsing and Normalization: Collected logs are automatically processed to become standardized in one format, which can be easily searched, categorized, and linked the events in an efficient manner.
- Real-Time Notification and Alerting Module: A rules engine and streaming platform are monitors that have a real- time look into classified logs. Important incidents or deviations cause an immediate alert to administrators through email or SMS messages or dashboard alerts.
- Scalable and Secure Backend Infrastructure: The system is based on a microservices design and implemented with the help of container platforms such as Docker and Kubernetes, which makes it scalable and modular. Secure APIs are used to control the data flow and sensitive logs are secured using JWT-based authentication.

Dashboard and Reporting Interface: Web Dashboard The web dashboard allows administrators to get clear information about the system status, log trends, and history of alerts. It facilitates dynamic querying, report creation, and setting up notification preferences enabling active monitoring and decision-making.

V. Results and Evaluation

The log management server was tested in terms of its performance and effectiveness in actual e-governance situations.

Key metrics included:

Log Ingestion Rate: 10 logs per second sustained without data loss.

Accuracy in Classification: 91.8% Content based log classification accuracy.

Alert Delivery Time: The mean time to issue alert is 1.5 seconds.

System Uptime: The availability of a 30-day continuous service test is 99.95% available.

User Satisfaction (through administrator survey): 89.2 Via survey of the users it was found that 89.2 percent said that there was better incident response and system visibility.

False Positive Rate: The false positives triggered were less than 3.5 percent of the overall alerts triggered.

The administrators reported quicker issue resolution, increased operational awareness and decreased manual effort in the log analysis. The fact that the system can scale and provide real-time notifications was particularly useful in the peak loads as well as in the environment where the logs are complex and multi-source.

VI. Advantages of Proposed System

- Categorizing intelligent Log Classification Machine learning models classify logs automatically based on their content and severity to enable progressive priority issue identification and lessen human analysis.
- Real-Time Event Detection and Notification The system continuously monitors incoming logs and sends timely alerts via email, SMS, or dashboard notifications, ensuring rapid response to critical events.
- 24/7 Availability and Scalability Built with a microservices architecture and containerization technologies like Docker and Kubernetes; the system guarantees high availability and can scale dynamically to handle peak workloads.
- Enhanced Administrator Experience With the intuitive dashboard gives administrators clear visibility into log trends, system health, and alert history, making monitoring more accurate and troubleshooting much more efficient.
- Domain Adaptability and Extensibility The system's modular design allows easy customization of classification models, notification rules, and integrations for different departments or e-governance services without affecting the entire system.
- Security and Compliance Strong authentication methods, including JWT and role-based access control, safeguard sensitive log data and ensure compliance with regulations.

VII. Conclusion

Looking ahead, there are several opportunities for enhancement. These include expanding multilingual log analysis to support diverse regional languages, improving anomaly detection with deep learning for greater accuracy, and incorporating offline or edge-device log processing to reduce latency. Future developments also envision voice- activated monitoring for hands-free administration and continuous real-time feedback loops to refine machine learning models based on operational data.

With these advancements, the system can evolve into an intelligent, adaptive solution that provides proactive insights and strengthens the digital governance infrastructure.

Future Scope

As promising as the current system is, there are several exciting directions for future development:

- Better Tracking with Open Telemetry In the future, we will add Open Telemetry to our system. This tool connects logs from different servers, allowing us to track a single user request from start to finish. This gives us a complete picture of how data moves through our microservices.
- Multilingual Capabilities Expanding language support would make the system usable in regions with diverse linguistic populations, increasing its global reach and applicability.
- Offline Functionality Adding offline capabilities with local data storage and lightweight NLP models would ensure the system remains operational even in areas with limited or unreliable internet connectivity.
- Integration into Smart Devices and IoT Embedding the system into smart home devices, kiosks, or wearables would allow users to interact with their environment in smarter, more intuitive ways.

Advanced Analytics for Stakeholders Future versions could offer dashboards and reports that help organizations track usage patterns, identify recurring issues, and make data- driven decisions to improve services.

References

[1] Belglazov, Y. Abduvaliyev, and R. Buyya, "Monitoring and Logging in Cloud Computing: Architecture, Taxonomy, and Future Directions," Journal of Cloud Computing, vol. 8, no. 1, pp. 1–25, 2019.

[2] J. D. Ullah, L. Gordon, and C. Phillips, "A Scalable Centralized Log Management System for Intrusion Detection in Large-Scale Networks," Journal of Network and Computer Applications, vol. 146, p. 102422, 2020.

[3] Y. He, F. Wu, and J. Camacho, "Machine Learning for Log Management: Methods and Challenges," ACM Computing Surveys, vol. 54, no. 1, pp. 1–30, 2022.

[4] Splunk Inc., "Best Practices in Centralized Log Management for Security and Compliance," White Paper, 2021. [Online]. Available: https://www.splunk.com. [Accessed: Jan. 15, 2026].

[5] New Relic, "2022 State of Logs Report," 2022. [Online]. Available: https://newrelic.com/resources/report. [Accessed: Jan. 15, 2026].

[6] Dynatrace, "Why Log Monitoring and Log Analytics Matter in a Hyperscale World," Dynatrace Blog, 2024. [Online]. Available: https://www.dynatrace.com/news/blog. [Accessed: Jan. 15, 2026].

[7] J. Alvarez and R. Singh, "Real-Time Streaming Log Analytics with Apache Kafka and Spark," in Proc. IEEE Big Data Conf., 2024, pp. 455–464.

[8] Y. Zhou, H. Li, and J. Wang, "Leveraging Large Language Models and BERT for Log Parsing and Anomaly Detection," Mathematics, vol. 12, no. 17, p. 2758, 2024.

[9] L. Palma et al., "Leveraging Large Language Models for Scalable and Explainable Cybersecurity Log Analysis," AI, vol. 5, no. 3, p. 55, 2025.

[10] S. Guan et al., "LogLLM: Log-based Anomaly Detection Using Large Language Models," arXiv preprint arXiv:2411.085, 2024.

[11] C. Yamini et al., "AI-Powered Server Log Management System Using Random Forest," Journal of Intelligent Systems, vol. 12, no. 3, pp. 145–158, 2023.